

**TRANSIMS:
TRansportation ANalysis SIMulation System**

Version: TRANSIMS-LANL-3.1

VOLUME SIX—INSTALLATION

31 August 2004

LA-UR-04-5849

Los Alamos
NATIONAL LABORATORY

Copyright, 2004, The Regents of the University of California. This software was produced under a U.S. Government contract (W-7405-ENG-36) by Los Alamos National Laboratory, which is operated by the University of California for the U.S. Department of Energy. The U.S. Government is licensed to use, reproduce, and distribute this software. Neither the Government nor the University makes any warranty, express or implied, or assumes any liability or responsibility for the use of this software.

VOLUME SIX—INSTALLATION

31 August 2004

The following individuals contributed to this document:

C. L. Barrett*
R. J. Beckman*
K. P. Berkbighler*
K. R. Bisset*
B. W. Bush*
S. Eubank*
K. M. Henson*
J. M. Hurford*
D. A. Kubicek*
M. V. Marathe*
P. R. Romero*
J. P. Smith*
L. L. Smith*
P. L. Speckman**
P. E. Stretz*
G. L. Thayer*
E. Van Eeckhout*
M. D. Williams*

* Los Alamos National Laboratory, Los Alamos, NM 87545

** National Institute of Statistical Sciences, Research Triangle Park, NC

Acknowledgments

This work was supported by the U. S. Department of Transportation (Assistant Secretary for Transportation Policy, Federal Highway Administration, Federal Transit Administration), the U.S. Environmental Protection Agency, and the U.S. Department of Energy as part of the Travel Model Improvement Program.

Volume Six: Contents

1. WHAT'S ON THE CD-ROM?	2
1.1 DISTRIBUTION DESCRIPTION.....	2
2. SYSTEM REQUIREMENTS.....	5
2.1 HARDWARE REQUIREMENTS.....	5
2.2 SOFTWARE REQUIREMENTS	6
3. INSTALLING TRANSIMS 3.1	8
3.1 INSTALLATION INSTRUCTIONS.....	8
3.2 BUILDING AND INSTALLING THE TRANSIMS COMPONENTS	9
3.3 RUNNING TRANSIMS USING THE DISTRIBUTION SCENARIOS.....	12
VOLUME SIX: INDEX	13

Volume Six: Figures

<i>Fig. 1. Directory structure found on the TRANSIMS 3.1 CD-ROM.</i>	<i>2</i>
---	----------

Volume Six: Tables

Table 1. Software—World Wide Web sites.....	7
--	----------

1. WHAT'S ON THE CD-ROM?

The TRANSIMS 3.1 distribution found on the CD-ROM contains the directory structure shown in Fig. 1. Once installed, the distribution's root directory is referred to as *TRANSIMS_HOME*. The CD-ROM also contains the install script, *install.sh*, and the release notes, *ReleaseNotes*.

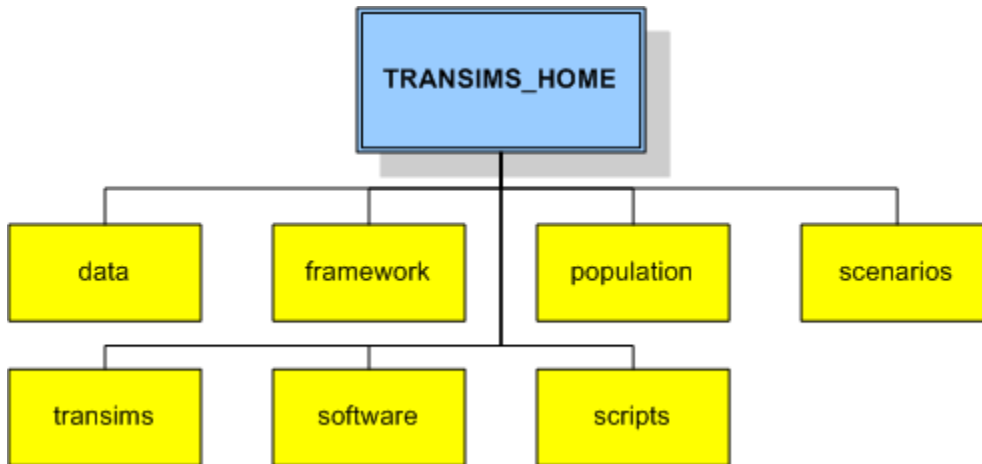


Fig. 1. Directory structure found on the TRANSIMS 3.1 CD-ROM.

1.1 Distribution Description

1.1.1 *software* Directory

The *software* directory contains distributions of the third-party software used by TRANSIMS.

1.1.2 *framework* Directory

The *framework* directory contains supporting software for the TRANSIMS Population Synthesizer.

1.1.3 *population* Directory

The *population* directory contains the build system and source code for the revised Population Synthesizer.

1.1.4 *scenarios* Directory

The *scenarios* directory (see Fig 2) contains subdirectories of scenarios to run the TRANSIMS 3.1 components. Three scenarios are included: *bignet*, *calibration*, and *gensig*.

- ***bignet*** – A multimode scenario and network used to run the TRANSIMS 3.1 components. The network contains ~16000 links.
- ***calibration*** – The microsimulation calibrations, which consist of one- and three-lane freeways, two-lane left turn, two-lane merge, and tee intersection networks.
- ***gensig*** – Creates generic signals for calibration and testing using the generic signal builder.

1.1.5 *scripts* Directory

The *scripts* directory contains scripts used to process data or control TRANSIMS 3.1 modules.

1.1.6 *transims* Directory

The *transims* directory contains the build system and source code for the TRANSIMS modules. The *source* subdirectory contains the following source codes for the TRANSIMS 3.0 components.

- *ACT* – TRANSIMS Activity Generator
- *ACTL* – Simplified Activity Generator
- *CA, PAR, TBX* – Traffic Microsimulator
- *CALIB* – Traffic Microsimulator calibration output filters
- *ENV* – Emissions Estimator
- *GBL* – Global definitions and methods used by other TRANSIMS components
- *IO* – TRANSIMS file interface definitions and methods
- *ITDB* – Iteration Database
- *NET* – TRANSIMS Network
- *OUT* – TRANSIMS Output
- *PLAN* – TRANSIMS route plan definition and methods
- *POPL* – Simplified Population Generator
- *ROUTER, THREADS* – Route Planner
- *SEL* – TRANSIMS Collator/Stratifier/Selector/Iteration Database
- *SYNPOP* – 1990 Population Synthesizer
- *TRANSIT* – Transit methods
- *TRAV* – Traveler definitions
- *VEH* – TRANSIMS Vehicle file generator
- *VIS2, mui* – Output Visualization

1.1.7 *data* Directory

The *data* directory contains information used by the 1990 Population Synthesizer and the Emissions Estimator.

2. SYSTEM REQUIREMENTS

2.1 Hardware Requirements

This version of TRANSIMS-LANL was tested on a Linux cluster with 60 dual processor nodes with two gigabytes of memory per node. Each processor is a 1.4 GHz Pentium III. Some scenarios were tested on single CPU Linux systems. Memory and disk requirements depend on the scenario that is used. All of the sample scenarios provided in this distribution are relatively small and can be run on a single CPU machine. Larger scenarios will require the parallel processing power of a Linux cluster.

2.1.1 Memory Requirements

- The *gensig* and *calibration* scenarios use less than 120 megabytes.
- The *bignet* scenario needs at least 512 megabytes.
- Memory requirements for the Output Visualizer depend on the size of the data files and are controllable by the user when the Visualizer is run.

2.1.2 Disk Requirements

The TRANSIMS distribution requires 1.6 gigabytes of disk space in the installation directory.

For the *calibration*, *gensig*, and *bignet* scenarios, five gigabytes of disk space support the execution and data collection for TRANSIMS 3.1 components using the output specifications in the TRANSIMS 3.1 configuration files supplied with the distribution. If the user increases output collection, additional disk space may be required. Operating System

2.1.3 Operating System

TRANSIMS-LANL was tested in a Linux cluster environment on RedHat Linux 8.0 and compiled with gcc/g++ version 3.3.2

Limited tests in a single CPU environment were done on RedHat Linux 9.0 using gcc/g++ 3.2.2. The *bignet* and *calibration* scenarios were compiled and tested in the single CPU environment.

2.1.3.1 Linux

To run the Traffic Microsimulator which uses MPI, or MPI, the Linux kernel must be compiled with networking support and must have assigned an IP address and a host name.

An actual network card is not required. The following options must be selected in the Linux kernel configuration:

- networking support (*CONFIG_NET*),
- System V IPC (*CONFIG_SYSVIPC*),
- TCP/IP networking (*CONFIG_INET*),
- dummy-net driver support (*CONFIG_DUMMY*), or
- the appropriate network card driver.

The default kernel shipped with Red Hat 8.0 and 9.0 is configured with the appropriate options. The following package categories should be selected during Red Hat Linux installation to run the TRANSIMS 3.1 components:

- X Window System,
- Mesa/GL, and
- Glut.

Additional package categories should be selected to compile the TRANSIMS 3.1 components:

- C Development,
- Development Libraries,
- C++ Development, and
- X Development.

2.2 Software Requirements

The TRANSIMS 3.1 distribution requires that the user install the following software.

2.2.1 Linux

- Autotools for configuring and building: *automake*, version 1.7 or higher; *libtool*, version 1.4.3 or higher; and *autoconf*, version 2.57 or higher.
- Compiler – gcc/g++ version 3.2.2 or higher
- X11R6 libraries (*Xmu*, *Xi*, *X11*, *Xext*, *Xt*)
- OpenGL and the OpenGL Utilities Toolkit libraries (*Mesa*)
- Linux libraries (*stdc++*, *ld-linux*, *ICE*, *SM*)
- Perl

Most of the third-party software used by TRANSIMS 3.1 is available on Red Hat Linux distribution CD-ROMs. The latest versions of the following packages should be installed: kernel, kernel-headers, gcc, glibc, libstdc++, make, perl, XFree86, Mesa, Mesa-devel, MPI, and Pvm. Precompiled versions of the random number generator, *sprng*, the graph partitioning software, *metis*, the logging system software, *log4cpp*, the test suite software, *cppunit*, the wxGTK toolkit used by the Visualizer, *wxGTK*, and *mpich* are supplied with the distribution in the software directory. The source code for the distributed software is in *software/src*.

The third-party software that can be obtained from the World Wide Web sites is listed in Table 1.

Table 1. Software—World Wide Web sites.

Software	World Wide Web site
Gnu C++ compiler	www.gnu.org/software/gcc/gcc.html
OpenGL/Mesa	http://www.mesa3d.org
PVM	http://www.epm.ornl.gov/pvm/pvm_home.html
MPI	http://www-unix.mcs.anl.gov/mpi/mpich
Perl	http://language.perl.com

3. INSTALLING TRANSIMS 3.1

TRANSIMS 3.1 requires that the operating system be installed with the software described in Section 2. The TRANSIMS 3.1 distribution uses 1.8 gigabytes of disk space.

3.1 Installation Instructions

We have divided the installation process into seven steps.

- Step One**
- Select a directory in which the distribution will reside. In these instructions, we will refer to this directory as *TRANSIMS_HOME*.
 - TRANSIMS users must have read and write permissions in this directory.

- Step Two**
- Mount/open the CD-ROM.

Linux:

- This operation may require root permissions.
- On Red Hat Linux, the CD-ROM directory is */mnt/cdrom*. This directory may be different on other Linux distributions.
- The directory in which the CD-ROM is mounted will be referred to as *CD_ROM_DIRECTORY*:

```
% /bin/mount <CD_ROM_DIRECTORY>
```

Example:

```
% /bin/mount /mnt/cdrom
```

- Step Three**
- The installation script copies the TRANSIMS 3.1 distribution from the CD-ROM to the *TRANSIMS_HOME* directory and installs it.
 - Run the installation script on the CD-ROM by using the following command:

```
% /bin/sh <CD_ROM_DIRECTORY>/install.sh <CD_ROM_DIRECTORY> <TRANSIMS_HOME>
```

Example:

Linux:

```
% /bin/sh /mnt/cdrom/install.sh /mnt/cdrom /home/transims
```

- Step Four** • Make sure that you have read and write permissions on the *TRANSIMS_HOME* directory and subdirectories.
- If necessary, change the file permissions in the TRANSIMS 3.1 distribution.
- Step Five** • Set the environment variable *TRANSIMS_HOME*. This is required in order to build TRANSIMS and also for all users who will be running the TRANSIMS 3.1 components.
- Bash Shell: Add the following line to your *.profile*

```
% export TRANSIMS_HOME=<full path name of directory where TRANSIMS is installed>
```
 - C-Shell: Add the following line to your *.cshrc*

```
% setenv TRANSIMS_HOME <full path name of directory where TRANSIMS is installed>
```
- Step Six** • Compile and install the TRANSIMS distribution using the instructions in Section 3.2. The *TRANSIMS_HOME/bin* directory is created during the installation process, and the generated executables are placed in it.
- Step Seven** • Before starting the Output Visualizer program (*Vis*) make sure that you are running X Windows.

Linux:

- To start the X server, use the *startx* command
(*/usr/X11R6/bin/startx*).

3.2 Building and Installing the TRANSIMS Components

3.2.1 The Build System

This release uses *autoconf*, *libtool*, and *automake* to configure and compile the source code. Required minimum versions of these tools are: *automake* - 1.7, *libtool* - 1.4.3, *autoconf* - 2.57.

3.2.2 Compilation Requirements

The TRANSIMS 3.1 distribution contains the source code for the TRANSIMS modules. Read the release notes text file, *ReleaseNotes*, on the CD-ROM. It contains important information about the compilation process.

The following software must be installed to compile TRANSIMS 3.1:

- Autotools: *automake* 1.7 or higher, *autoconf* 2.57 or higher, *libtool* 1.4.3 or higher
- Compiler—Gnu C++ (gcc 3.2.2 or higher).
- OpenGL/Mesa Version 3.0 or higher
- X11R6 (include files and libraries)
- *metis*, *pvm*, *mpi*, and *sprng* from TRANSIMS distribution or the equivalent versions
- Gnu make
- MPI must be configured, built, and installed on your system. The TRANSIMS distribution contains an *mpich* distribution of MPI which must be configured, built, and installed prior to compiling TRANSIMS. See the MPI distribution documents for instructions on configuring, building, and running MPI. Use the configuration options `-disable-f77`, `-disable-f90`, and `-disable-f90modules` to eliminate building the FORTRAN portions of *mpich*, which TRANSIMS does not use. Running an example program such as *cpi* from the *mpich* distribution is strongly suggested to test whether MPI works correctly on your machine configuration. If your system already has an MPI installation, edit the *build.sh* script below in Step Three to point to the appropriate MPI root directory.

3.2.2.1 Compilation Instructions

Step One Install the appropriate software as described above and set the environment variable `TRANSIMS_HOME` to point to the root directory of the TRANSIMS 3.1 distribution.

Step Two Change directory to `TRANSIMS_HOME/transims`. Create a subdirectory `TRANSIMS_HOME/transims/build`. The TRANSIMS distribution will be configured and built in the *build* directory.

See the Release Notes supplied with the distribution for important compilation instructions specific to this release.

Step Three Copy the build script `TRANSIMS_HOME/transims/source/build.sh` to the build directory.

```
% cd $TRANSIMS_HOME/transims/build
% cp $TRANSIMS_HOME/transims/source/build.sh .
```

The *build.sh* script defines the locations of software packages and the compiler to use during the build process. The script points to the TRANSIMS distributions as the default location for software required for compilation. Edit the script to point to your system installations if you wish to use them. The default installation directory (`INSTALL_DIR`) is `TRANSIMS_HOME/bin`. Change the definition in the *build.sh* script to another directory if desired.

TRANSIMS should be compiled with gcc/g++ version 3.2 or higher. The default location of the compiler in the *build.sh* script is */usr/bin*. Edit the *build.sh* script to specify a different location for the compiler(CC,CXX).

Run the *build.sh* script to configure and create the necessary makefiles.

```
% ./build.sh
```

Type *make* in the *build* directory to compile the TRANSIMS distribution

```
% make
```

Step Four Type *make install* in the *build* directory to install the executables in the install directory, *TRANSIMS_HOME/bin*, and the libraries in *TRANSIMS_HOME/lib*.

```
% make install
```

Step Five Compiling the Population Synthesizer

Change directory to *TRANSIMS_HOME/framework*. Create the directory *TRANSIMS_HOME/framework/build*. Copy the *build.sh* script from *TRANSIMS_HOME/framework/src* to the *build* directory.

```
% mkdir $TRANSIMS_HOME/framework/build
% cd $TRANSIMS_HOME/framework/build
% cp $TRANSIMS_HOME/framework/src/build.sh
```

The script points to the TRANSIMS distributions as the default location for software required for compilation. Edit the script to point to your system installations if you wish to use them.

Run the *build.sh* script to configure and create the necessary makefiles.

```
% ./build.sh
```

Type *make* in the *build* directory to compile the framework library

```
% make
```

Type *make install* in the *build* directory to install the library, *libUISlib.a*, in *TRANSIMS_HOME/lib*

```
% make install
```

Change directory to *TRANSIMS_HOME/population*. Create the *build* directory, *TRANSIMS_HOME/population/build*. Change directory to the *build* directory. Copy the *build.sh* script from *TRANSIMS_HOME/population/src* to the *build* directory.

```
% mkdir $TRANSIMS_HOME/population/build
% cd $TRANSIMS_HOME/population/build
% cp $TRANSIMS_HOME/population/src/build.sh
```

The script points to the TRANSIMS distributions as the default location for software required for compilation. Edit the script to point to your system installations if you wish to use them.

Run the *build.sh* script from the *TRANSIMS_HOME/population/build* directory to configure and create the necessary makefiles.

```
% ./build.sh
```

Type `make` in the *build* directory to compile the population synthesizer.

```
% make
```

Type `make install` in the *build* directory to install the executables (*ConformSummary*, *GeneratePopulation*, *IPF*, *CombineSummary.pl*, *EditGeocorr.pl*, *SplitPums.pl*, *CombinePums.pl*) in the install directory *TRANSIMS_HOME/bin*.

```
% make install
```

3.3 Running TRANSIMS Using the Distribution Scenarios

Three scenarios are supplied with the TRANSIMS distribution. The *bignet* scenario uses all of the TRANSIMS modules. *TRANSIMS_HOME/scenarios/bignet/scripts* contains scripts to run the modules. *RunAll* in this directory is a road map to the execution sequence of the modules.

The *calibration* scenarios demonstrate behavior of the Traffic Microsimulator on a series of roadway configurations, 1-lane freeway, 3-lane freeway, left turns against traffic, merging behavior, and turning behavior at a signalized intersection.

The *gensig* scenario demonstrates the generation and use of signalized traffic control parameters on a small test network.

Instructions on use of the Population Synthesizer are in the TRANSIMS documents, Volume Three, Chapter Two, *Population Generation*. The document describes an example that is included in the distribution in the *\$TRANSIMS_HOME/population/example* directory.

Complete instructions for use and execution of the TRANSIMS components in all of the scenarios are in Volume Four (*Calibrations, Scenarios, and Tutorials*) in the TRANSIMS document set available on <http://transims.tsasa.lanl.gov>.

Volume Six: Index

- ACT*, 3
- ACTL*, 3
- autoconf*, 6, 9, 10
- automake*, 6, 9, 10
- Bash Shell, 9
- bignet*, 3
- C Development, 6
- C++ Development, 6
- CA*, 3
- CALIB*, 3
- calibration*, 3
- CD-ROM, 2, 8
- Compilation Requirements**, 9
- CONFIG_DUMMY*, 6
- CONFIG_INET*, 6
- CONFIG_NET*, 6
- CONFIG_SYSVIPC*, 6
- C-Shell, 9
- data* Directory, 4
- Development Libraries, 6
- Disk Requirements**, 5
- Emissions Estimator, 3, 4
- ENV*, 3
- framework* Directory, 2
- GBL*, 3
- gcc, 7, 10
- gensig*, 3
- Glut, 6
- Gnu C++, 7, 10
- Gnu C++ compiler, 7
- Gnu make, 10
- Hardware Requirements**, 5
- ICE*, 7
- Installation Instructions**, 8
- IO*, 3
- ITDB*, 3
- Iteration Database, 3, 4
- kernel, 5, 6
- ld-linux*, 7
- libtool*, 6, 9, 10
- Linux, 5, 6, 7, 8, 9
- Linux cluster, 5
- Linux libraries, 7
- Memory requirements, 5
- Mesa, 6, 7
- Mesa/GL, 6
- metis*, 10
- mpi*, 7, 10
- MPI, 5, 7
- mui*, 4
- NET*, 3
- OpenGL Utilities Toolkit libraries, 7
- Operating system, 8
- OUT*, 3
- Output Visualization, 4
- PAR*, 3
- perl, 7
- Perl, 7
- PLAN*, 3
- POPL*, 3
- population* Directory, 2
- pvm*, 7, 10
- PVM, 5, 7
- ReleaseNotes*, 2, 9
- Route Planner, 3
- ROUTER*, 3
- scenarios* directory, 2
- scenarios* Directory, 2
- scripts* directory, 3
- SEL*, 4
- Simplified Activity Generator, 3
- Simplified Population Generator, 3
- SM*, 7
- software* Directory, 2
- Software Requirements**, 6
- source* directory, 3
- sprng*, 10
- stdc++*, 7
- SYNPOP*, 4
- SYSTEM REQUIREMENTS**, 5
- TBX*, 3
- THREADS*, 3
- Traffic Microsimulator, 3, 5, 12
- TRANSIMS Activity Generator, 3
- transims* Directory, 3
- TRANSIMS Network, 3
- TRANSIMS Output, 3
- TRANSIT*, 4
- TRAV*, 4
- VEH*, 4
- VIS*, 4
- X Development, 6
- X Window system, 6
- X11R6, 6, 9, 10
- X11R6 libraries, 6